DOCUMENT RESUME

ED 108 644 IR 002 165

AUTHOR Kline, Lanaii
TITLE Inventory of Assets II.
INSTITUTION Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
REPORT NO SWRL-TN-5-72-59
PUB DATE 17 Nov 72
NOTE 19p.; Pages 11-17 may reproduce poorly

EDRS PRICE MF-$0.76 HC-$1.58 PLUS POSTAGE
DESCRIPTORS *Computer Programs; Data Processing; Design; Electronic Data Processing; *Equipment; *Facility Inventory; Flow Charts; Information Processing; Information Systems; *Management Information Systems; Recordkeeping
IDENTIFIERS Computer Software Documentation; Design Specifications; Equipment Inventory; *Inventory Of Assets II

ABSTRACT

INVENTORY OF ASSETS computer programs differ from the first version of INVENTORY OF ASSETS in that the sort routines used to order the input data are done using the IBM SORT program. This method conserves nearly 700,000 bytes of core storage, thereby decreasing the time required to receive processed jobs by 24 hours. However, this method restricts the program to IBM 360 computers. This document is intended to serve as supplemental software documentation and should be read in conjunction with IR 002 163. (Author/DGC)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED103644

IR

SWRL

SOUTHWEST REGIONAL LABORATORY TECHNICAL NOTE

DATE: November 17, 1972

NO: TN 5-72-59

TITLE: INVENTORY OF ASSETS II

AUTHOR: Lanaii Kline

ABSTRACT

This document describes the modifications to the Inventory of Assets Program which resulted in INVENTORY OF ASSETS II Program.

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

IR 002 165
14

This document is intended for internal staff distribution and use. Permission to reprint or quote from this working document, wholly or in part, should be obtained from SWRL, 4665 Lampson Avenue, Los Alamitos, California 90720

INVENTORY OF ASSETS II

Lanaii Kline

I. Introduction

INVENTORY OF ASSETS II* differs from INVENTORY OF ASSETS in that the sorts are done using IBM SORT and are read in as data to the program. This method conserves nearly 700,000 bytes of core, thereby, decreasing the turnaround time by twenty-four hours. This method as described below restricts the program to IBM 360's.

II. Description

The documentation for INVENTORY OF ASSETS (TN 5-72-56) will also serve as documentation to this program with the changes as noted below:

This program does not require the use of the functions SORT or COMPAR. Therefore, the arrays LIST and IHEAD are not needed, as well as, the integer COMPAR. Since there is no need to store the input for in-core sorting; the real array PRICE is declared as a real scalar, the integer arrays SIGN and AMP are declared to be integer scalars, and the integer array MORE is declared to be a singly dimensioned array of thirty-five elements.

The points at which the arrays were sorted in INVENTORY OF ASSETS are now points at which the appropriate file is read.

Attachment 1, Program Flowchart, and Attachment 2, Program Listing, provide a more detailed description of the program logic. These attachments should be substituted for sections 8.0 and 9.0 of the INVENTORY OF ASSETS documentation when TN 5-72-36 is being used as a descriptor for this program

III. Preparation of the Database

The data for the inventory of assets reports must be updated each quarter using the update program. Before the update program is begun, check for the following errors:

1. The updates must be arranged in a numerically ascending order on columns one through five. Therefore, local id 4380 must precede local id 4388.

---

* This is a modification of the Inventory of Assets Program documented in TN 5-72-56 by Lanaii Kline.

2. All change (c in column 79) update cards must contain all of the data currently on the record as well as the change being made, i.e.,

current record

Col 1	Col 42	Col 7980
2140-01CHAIR	47.50	1
2140-01ARMCHAIR		2

change record:

Col 1	Col 42	Col 7980
2140-01CHAIR	48.50	C 1
2140-01ARMCHAIR		C 2

where the only change made is on the unit price.

3. All update cards must have a N, C, or D punched in column 79. N is for new or insertion; C is for change; and D is for deletion.

4. An update with two cards for a given local id must have the "one" card before the "two" card (column 80).

Example (update input cards)

Col 1	Col 42	Col 7980
123-04CHAIR	27.50	N 1
123-04ARMCHAIR		N 2
4754-16DESK	28.72	C 1
4754-16EXECUTIVE		C 2
4755-14TABLE	127.53	C 1
4755-14CONFERENCE		C 2
4756-16		D 1

(See Attachment 3)

Job Set-up:

Col 1

```
// Job Card
// Password Card
// EXEC FORTGCLG
//FORT.SYSIN DD *
      program deck (See TN 5-72-52)
```

```
/*
//GO.FT03F001 DD DSN=AGB005.SWR.K.In[1],DISP=(OLD,KEEP),UNIT=2314,
// VOL=SER=SWRL01,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200)
//GO.FT04F001 DD DSN=AGB005.SWR.K.Im[1]DISP=(OLD,KEEP),UNIT=2314,
// VOL=SER=SWRL01,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200)
//GO.SYSIN DD *
     update cards
/*
//
```

The output file will be the input file for the sort steps (//SORT.SORTIN) and the input file (//GO.FT01F001) for the report run.

See Section 6.6 of TN 5-72-52 for error messages generated by the update program. Repeat this update step changing the input and output dataset names after correcting the invalid update cards. Use only the rejected updates as input. For example, if in the example of update input cards the card with local id 4754 generated the message:

FOUND NO MATCHING LOCAL ID FOR THE CHANGE LOCAL ID 4754,

then find the cause of the error, (local id should have been 4753). Change the card and resubmit with

```
Col 1                        Col 40                      Col 7980

4753-16DESK                      28.73                        C 1
4753-16EXECUTIVE                                              C 2
```

as input only.

A. Sorted on Category number

```
Col 1

// Job Card
// Password Card
// EXEC SORTA,SPACE='(50,50)'
//SORT.SORTIN DD DSN=AGB005.SWR.K.Im[1],DISP=(SHR,KEEP),UNIT=2314,
// VOL=SER=SWRL01,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200)
//SORT.SORTOUT DD DSN=AGB005.SWR.K.I2,DISP=(NEW,KEEP),UNIT=2314,
// VOL=SER=DATE02,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
// SPACE=(TRK,(50,50),RLSE)
//SORT.SYSIN DD *
   SORT FIELDS=(7,2,CH,A),SIZE=E6000
   END
/*
//
```

1. If m equals ONE, then n equals 1. If m equals 1, then n equals ONE.

B. Sorted on Check Number

Col 1

```
        :
        :
//SORT.SORTOUT DD DSN=AGB005.SWR.K.I3,DISP=(NEW,KEEP),UNIT=2314,
// VOL=SER=DATE03,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
// SPACE=(TRK,(50,50),RLSE)
//SORT.SYSIN DD *
   SORT FIELDS=(47,5,CH,A),SIZE=E6000
   END
/*
//
```

C. Sorted on Contract Number and Local ID

Col 1

```
        :
        :
//SORT.SORTOUT DD DSN=AGB005.SWR.K.I4,DISP=(NEW,KEEP),UNIT=2314,
// VOL=SER=DATE04,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
// SPACE=(TRK,(50,50),RLSE)
//SORT.SYSIN DD *
   SORT FIELDS=(26,4,CH,A,1,5,CH,A),SIZE=E6000
   END
/*
//
```

D. Sorted on Contract Number and Category

Col 1

```
        :
        :
//SORT.SORTOUT DD DSN=AGB005.SWR.K.I5,DISP=(NEW,KEEP),UNIT=2314,
// VOL=SER=DATE05,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
// SPACE=(TRK,(50,50),RLSE)
//SORT.SYSIN DD *
   SORT FIELDS=(26,4,CH,A,7,2,CH,A),SIZE=E6000
   END
/*
//
```

E. Sorted on Contract Number and Check Number

Col 1

```
        :
        :
```

```
//SORT.SORTOUT DD DSN=AGB005.SWR.K.I6,DISP=(NEW,KEEP),UNIT=2314,
// VOL=SER=DATE06,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
// SPACE=(TRK,(50,50),RLSE)
//SORT.SYSIN DD *
   SORT FIELDS=(26,4,CH,A,47,5,CH,A),SIZE=E6000
   END
/*
//
```

IV. Production Run

When the sorts are completed run the inventory with the new version as follows:

Col 1

```
// Job Card
// Password Card
// EXEC FORTGCLG
//FORT.SYSIN DD *
        .
        .
      program deck
        .
        .
/*
//GO.FT06F001 DD SYSOUT=TEXT2,COPIES=3,
// UNIT=(SYSDA,4),SPACE=(TRK,(100,50),RLSE)
//GO.FT01F001 DD DSN=AGB005.SWR.K.Im1,UNIT=2314,DISP=(OLD,KEEP),
// VOL=SER=SWRL01,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200)
//GO.FT02F001 DD DSN=AGB005.SWR.K.I2,UNIT=2314,DISP=(OLD,KEEP),
// VOL=SER=DATE02,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200)
//GO.FT03F001 DD DSN=AGB005.SWR.K.I3,UNIT=2314,DISP=(OLD,KEEP),
// VOL=SER=DATE03,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
//GO.FT04F001 DD DSN=AGB005.SWR.K.I4,UNIT=2314,DISP=(OLD,KEEP),
// VOL=SER=DATE04,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
//GO.FT07F001 DD DSN=AGB005.SWR.K.I5,UNIT=2314,DISP=(OLD,KEEP),
// VOL=SER=DATE05,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200),
//GO.FT08F001 DD DSN=AGB005.SWR.K.I6,UNIT=2314,DISP=(OLD,KEEP),
// VOL=SER=DATE06,DCB=(RECFM=FB,LRECL=160,BLKSIZE=7200)
//GO.SYSIN DD *
  Date Card
/*
//
```

V. Deletion of Temporary Sort File

Once the output has been improved, delete the sort files as fol ows:

Col 1

```
// Job Card
// Password Card
// EXEC PGM=IEFBR14
//OUT1 DD DSN=AGB005.SWR.K.I2,UNIT=2314,DISP=(OLD,DELETE),
// VOL=SER=DATE02
//OUT2 DD DSN=AGB005.SWR.K.I3,UNIT=2314,DISP=(OLD,DELETE),
// VOL=SER=DATE03
//OUT3 DD DSN=AGB005.SWR.K.I4,UNIT=2314,DISP=(OLD,DELETE),
// VOL=SER=DATE04
//OUT4 DD DSNAME=AGB005.SWR.K.I5,UNIT=2314,DISP=(OLD,DELETE),
// VOL=SER=DATE05
//OUT5 DD DSNAME=AGB005.SWR.K.I6,UNIT=2314,DISP=(OLD,DELETE),
// VOL=SER=DATE06
//
```

VI. System Completion Codes

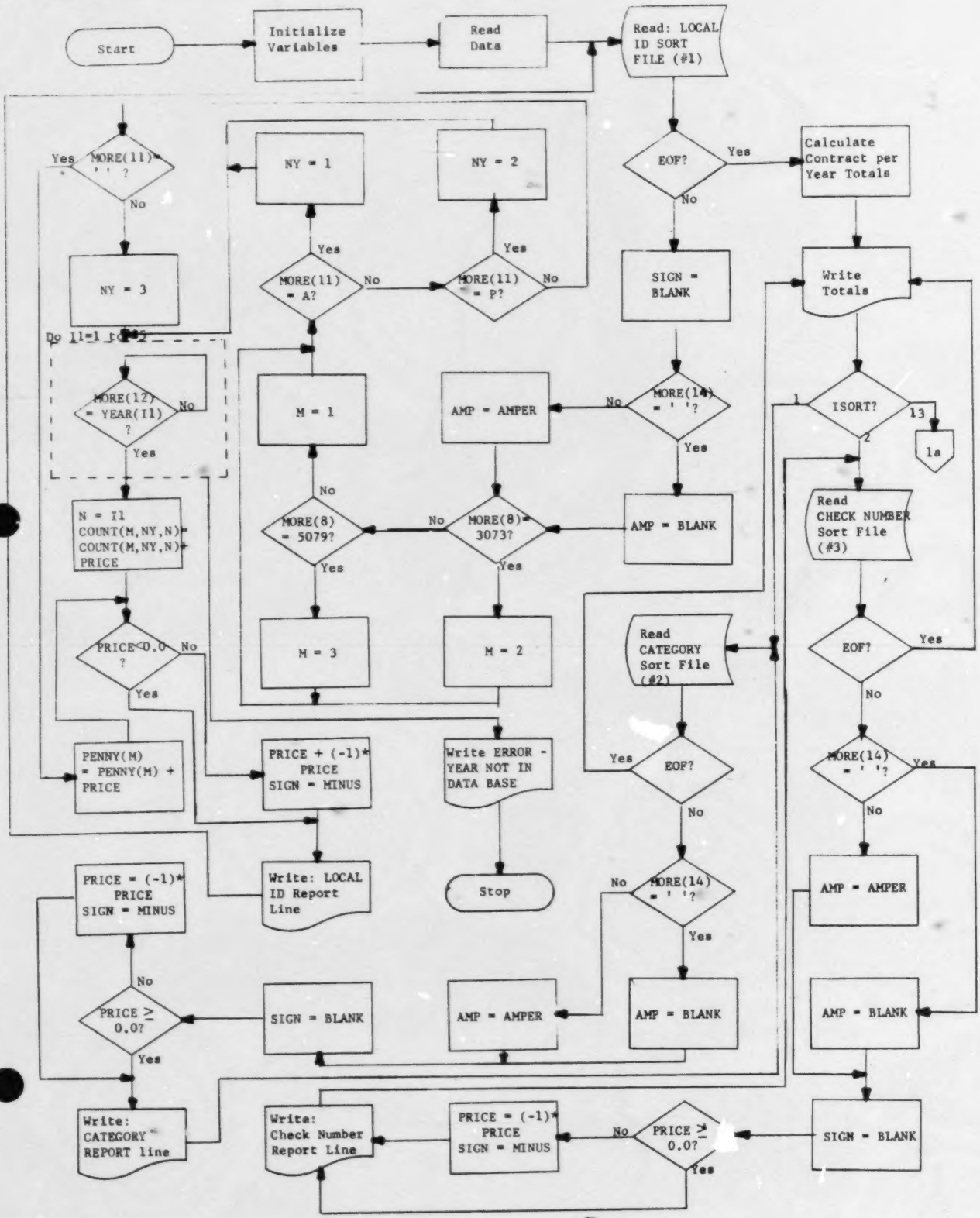
Code	Meaning	Action
80A	Ran out of core space	Resubmit job after adding RG or RC = nnnk*
322	Ran out of time	Resubmit job after increasing the time estimate on the job card (#s)
E22	Ran out of I/O requests	Rerun after increasing I/O request on the job card (####I)
D37	Ran out of paper	Rerun after increasing the space parameter on the GO.FT06F001 card

* RC - if the 80A was on the compile step
RG - if the 80A was on the execute step

SWRL

Flowchart Layout Form

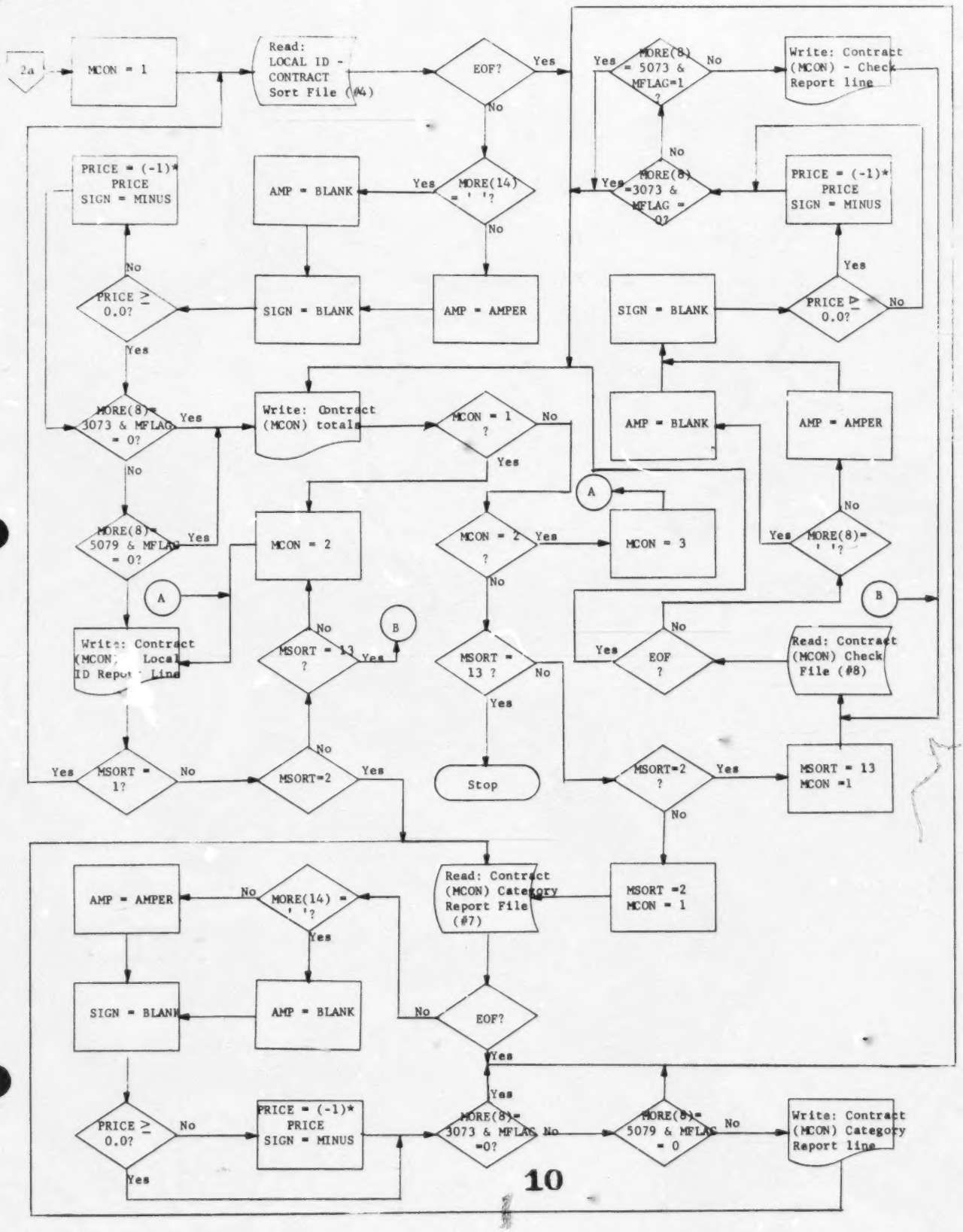
Programmer: Lanaii Kline
Program ID: INVENTORY OF ASSETS II
Date: Oct. 31, 1972
Page 1 of 2



SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanali Kline
PROGRAM ID: INVENTORY OF ASSETS II
DATE: Oct. 31, 1972
PAGE 2 OF 2



9.0 - PROGRAM LISTING

```
C**********************************************************************
C
C     MORE(J),J=1,35
C           MORE(1)              : LOCAL ID
C           MORE(2)              : CATEGORY
C           MORE(3)-MORE(7)      : ARTICLE
C           MORE(8)              : CONTRACT NUMBER-FIRST PORTION
C           MORE(9)-MORE(10)     : CONTRACT NUMBER
C           MORE(11)             : ADM-PROG-SP
C           MORE(12)             : BUDGET YEAR
C           PRICE                : PRICE
C           MORE(13)-MORE(15)    : CHECK NUMBER
C           MORE(16)-MORE(17)    : DATE
C           MORE(18)-MORE(21)    : SERIAL NUMBER
C           MORE(22)             : LOCAL ID
C           MORE(23)             : CATEGORY
C           MORE(24)-MORE(33)    : DESCRIPTION
C           MORE(34)-MORE(35)    : LOCATION
C
C**********************************************************************
      IMPLICIT INTEGER (A-Z)
      REAL*8 TOTAL(3,5),COUNT(3,3,5),PENNY(3),GTOTAL(3,4),GRAND(4),
     +DOLLAR,PRICE
      DIMENSION MORE(35),APS(3),YEAR(5),CON(3)
      COMMON /X1/PAGE
      COMMON /X2/ISORT,MSORT,NCOUNT,DATE(20)
      COMMON /X3/CON
      DATA TWO/'2'/,APS/'ADM ','PROG','SPEC'/,MINUS,BLANK/'-',' '/
      DATA A,P,S,YEAR/'A','P','S',0,59,70,71,72/,AMPER/'&'/
      MSORT=1
      DOLLAR=0.0
      DO 100 I=1,3
      DO 100 J=1,3
      DO 100 K=1,5
      TOTAL(I,K)=0.0
  100 COUNT(I,J,K)=0.0
      DO 4100 I=1,3
 4100 PENNY(I)=0.0
      DO 453 J=1,3
      DO 453 I=1,4
      GTOTAL(J,I)=0.0
  453 GRAND(I)=0.0
      I=1
      NCOUNT=0
      ISORT=1
      WRITE(6,10)
   10 FORMAT('1BEGIN PROCESSING')
      READ(5,9999) (DATE(I),I=1,20)
 9999 FORMAT(20A4)
      PAGE=1
      WRITE(6,60) (DATE(IDAY),IDAY=1,20),PAGE
   60 FORMAT('1',20X,20A4,20X,'PAGE',I3)
      WRITE(6,17)
   17 FORMAT('0',54X,'SORTED BY LOCAL ID NUMBER')
      WRITE(6,11)
   11 FORMAT('0')
      NCOUNT=6
 5666 READ(1,1,END=1333) (MORE(I),I=1,12),PRICE,(MORE(J),J=13,35)
    1 FORMAT(I5,1X,12A4,A1,I4,A4,2A1,I2,F9.2,I5,1X,2(A4,A1),3A4,
```

```
     +A2,4X,I5,1X,I2,9A4,A2,2A4)
      SIGN=BLANK
      IF(MORE(14).EQ.BLANK) GO TO 4002
      AMP=AMPER
4003  IF(CON(2).EQ.MORE(1)) GO TO 400
      IF(CON(3).EQ.MORE(1)) GO TO 423
C
C           CONTRACT # EQUALS CON(1)
C
      N=1
405   IF(MORE(11).EQ.A) GO TO 401
      IF(MORE(11).EQ.P) GO TO 402
      IF(MORE(11).EQ.BLANK) GO TO 4001
C
C           SPECIAL
C
      NY=3
406   DO 403 I1=1,5
      IF(MORE(12).EQ.YEAR(I1)) GO TO 404
403   CONTINUE
C
C           NO YEAR MATCH
C
      WRITE(6,1000)
1000  FORMAT(' ERROR - YEAR DOES NOT MATCH ANY IN ARRAY YEAR')
      GO TO 2100
C
C           THERE IS NOT ANOTHER CHECK NUMBER ON THIS CARD
C
4002  AMP=BLANK
      GO TO 4003
404   M=I1
      COUNT(N,NY,J)=COUNT(N,NY,J)+PRICE
      IF(PRICE.LT.0.0) GO TO 450
      GO TO 6
C
C           A CREDIT
C
450   PRICE=(-1)*PRICE
      SIGN=MINUS
      GO TO 6
C
C           CONTRACT # EQUALS CON(2)
C
400   N=2
      GO TO 405
C
C           CONTRACT # EQUALS CON(3)
C
423   N=3
      GO TO 405
C
C           NOT ADM., PROG., OR SPECIAL --- IT IS A BLANK
C
4001  PENNY(J)=PENNY(J)+PRICE
      IF(PRICE.LT.0.0) GO TO 450
      GO TO 6
C
```

```
C              ADMINISTRATIVE
C
401    KY=1
       GO TO 406
C
C              PROGRAMATIC
C
402    KY=2
       GO TO 405
C
C              SORT ON LOCAL ID
C
6      WRITE(6,9) (MORE(J),J=3,10),(MORE(K),K=1,2),(MORE(L),L=11,12),
      +PRICE,SIGN,MORE(13),AMP,(MORE(KK),KK=14,21),
      +(MORE(LL),LL=24,35)
9      FORMAT(' ',4A4,A1,1X,I4,A4,A1,1X,I5,'-',I2,2X,A1,2X,I2,1X,F9.2,
      +A1,1X,I5,A1,A4,A1,1X,A4,A1,1X,3A4,A2,1X,9A4,A1,1X,A4,A2)
       NCOUNT=NCOUNT+1
       IF(NCOUNT.GT.50) CALL NEWPAG(NCOUNT)
       GO TO 6666
C
C              SORTED BY CATEGORY NUMBER
C
3100   PAGE=1
       NCOUNT=0
       WRITE(6,50) (DATE(IDMY),IDMY=1,20),PAGE
       WRITE(6,16)
16     FORMAT('0',54X,'SORTED BY CATEGORY NUMBER')
       WRITE(6,11)
11     FORMAT('0',4X,'ARTICLE',7X,'CONTRACT',2X,'LOCAL ID',1X,'ADD',1X,
      +'YR',6X,'UNIT',2X,'CHECK',7X,'DATE',2X,'SERIAL NUMBER',2X,
      +'DESCRIPTION',25X,'LOCATION'/19X,'NUMBER',13X,'PRO',8X,'PRICE',
      +2X,'NUMBER'/38X,'SP'/'C')
       NCOUNT=NCOUNT+6
       ISORT=2
14     READ(2,1,END=13) (MORE(I),I=1,12),PRICE,(MORE(J),J=13,35)
       IF(MORE(14).EQ.BLANK) GO TO 5110
       AMP=AMPER
       GO TO 5111
5110   AMP=BLANK
5111   CONTINUE
       SIGN=BLANK
       IF(PRICE.GE.0.0) GO TO 9924
       PRICE=(-1)*PRICE
       SIGN=MINUS
9924   CONTINUE
       WRITE(6,9) (MORE(J),J=3,10),(MORE(K),K=1,2),(MORE(L),L=11,12),
      +PRICE,SIGN,MORE(13),AMP,(MORE(KK),KK=14,21),
      +(MORE(LL),LL=24,35)
       NCOUNT=NCOUNT+1
       IF(NCOUNT.GT.50) CALL NEWPAG(NCOUNT)
       GO TO 14
C
C              SORTED BY CHECK NUMBER
C
19     ISORT=13
       PAGE=1
       NCOUNT=0
```

```
      WRITE(6,60) (DATE(IDMY),IDMY=1,20),PAGE
     +WRITE(6,21)
21    FORMAT('0',56X,'SORTED BY CHECK NUMBER')
      WRITE(6,11)
      NCOUNT=NCOUNT+6
18    READ(5,1,END=13) (MORE(I),I=1,12),PRICE,(MORE(J),J=13,35),
      IF(MORE(14).EQ.BLANK) GO TO 5112
      AMP=AMPER
      GO TO 5113
5112  AMP=BLANK
5113  CONTINUE
      SIGN=BLANK
      IF(PRICE.GE.0.0) GO TO 9925
      PRICE=(-1)*PRICE
      SIGN='MINUS'
9925  CONTINUE
      WRITE(6,9) (MORE(J),J=3,10),(MORE(K),K=1,2),(MORE(L),L=11,12),
     +PRICE,SIGN,MORE(13),AMP,(MORE(KK),KK=14,21),
     +(MORE(LL),LL=24,35)
      NCOUNT=NCOUNT+1
      IF(NCOUNT.GT.50) CALL NEWPAG(NCOUNT)
      GO TO 18
1333  DO 4501 I=1,3
      GRAND(4)=GRAND(4)+PENNY(I)
      GTOTAL(I,4)=GTOTAL(I,4)+PENNY(I)
      DO 4501 K=1,5
      DO 4502 J=1,3
      GTOTAL(I,J)=GTOTAL(I,J)+COUNT(I,J,K)
      TOTAL(I,K)=TOTAL(I,K)+COUNT(I,J,K)
4502  GRAND(J)=GRAND(J)+COUNT(I,J,K)
      GTOTAL(I,4)=GTOTAL(I,4)+TOTAL(I,K)
4501  GRAND(4)=GRAND(4)+TOTAL(I,K)
      DOLLAR=PENNY(1)+PENNY(2)
13    WRITE(6,2101) (DATE(IDMY),IDMY=1,20)
2101  FORMAT('1'//'0',20X,20A4)
      DO 2108 I=1,3
      WRITE(6,2104) CON(I)
2104  FORMAT('0CONTRACT NUMBER',1X,I4//' ',35X,'TOTAL',4X,
     +'ADMINISTRATIVE',8X,'PROGRAM',5X,'SPECIAL PROGRAM',10X,'NONE OF
     +ESE'//'0')
      DO 2106 K=1,5
      L=YEAR(K)+1900
      IF(L.NE.1900) GO TO 2109
      IF(COUNT(I,1,K).EQ.0.AND.COUNT(I,2,K).EQ.0.AND.COUNT(I,3,K).EQ.
     +0) GO TO 2106
      WRITE(6,2110) TOTAL(I,K),(COUNT(I,J,K),J=1,3)
2110  FORMAT(' ',30X,F10.2,5X,F9.2,10X,F9.2,7X,F9.2)
      GO TO 2106
2109  IF(COUNT(I,1,K).EQ.0.AND.COUNT(I,2,K).EQ.0.AND.COUNT(I,3,K).EQ.
     +0) GO TO 2106
      WRITE(6,2103) L,TOTAL(I,K),(COUNT(I,J,K),J=1,3)
2103  FORMAT(' ',15X,I4,11X,F10.2,5X,F9.2,10X,F9.2,7X,F9.2)
2106  CONTINUE
      WRITE(6,2107) CON(I),GTOTAL(I,4),(GTOTAL(I,J),J=1,3),PENNY(I)
2107  FORMAT('0TOTAL CONTRACT ',I4,11X,2(F10.2,'*',3X),2(5X,F10.2,'*'
2108  CONTINUE
      WRITE(6,2004) GRAND(4),(GRAND(J),J=1,3),DOLLAR
2004  FORMAT('0GRAND TOTAL COMPANY',10X,2(F10.2,'**',3X),4X,F10.2,'**
```

```
     +2(9X,F10.2,'**'),43,F10.2,'**'/'1')
      IF(ISORT.EQ.1) GO TO 3100
      IF(ISORT.EQ.2) GO TO 19
C
C              SORT ON LOCAL ID NUMBER, PER CONTRACT NUMBER
C
      ISORT=8
      MCOUNT=0
      MFLAG=0
      PAGE=1
      NCON=1
      WRITE(5,60) (DATE(I),I=1,20),PAGE
      WRITE(6,177) CON(NCON)
177   FORMAT('0',41X,'SORTED BY  LOCAL ID NUMBER PER CONTRACT NUMBER',
     +1X,I4)
      WRITE(6,11)
      MCOUNT=8
      ISORT=1
144   READ(4,1,END=133) (MORE(I),I=1,12),PRICE,(MORE(J),J=13,35)
      IF(MORE(14).EQ.BLANK) GO TO 5116
      AMP=AMPER
      GO TO 5117
5116  AMP=BLANK
5117  CONTINUE
      SIGN=BLANK
      IF(PRICE.GE.0.0) GO TO 9920
      PRICE=(-1)*PRICE
      SIGN=MINUS
9920  CONTINUE
      IF((MORE(4).EQ.3073.AND.MFLAG.EQ.0).OR.(MORE(3).EQ.5079.AND.
     +MFLAG.EQ.1)) GO TO 133
130   WRITE(6,9) (MORE(J),J=3,10),(MORE(K),K=1,2),(MORE(L),L=11,12),
     +PRICE,SIGN,MORE(13),AMP,(MORE(KK),KK=14,21),
     +(MORE(LL),LL=24,35)
      MCOUNT=MCOUNT+1
      IF(MCOUNT.GT.50) CALL NEWPAG(MCOUNT)
      IF (MSORT .EQ. 1) GO TO 144
      IF (MSORT .EQ. 2) GO TO 156
      IF (MSORT .EQ. 13) GO TO 451
C
C              CHANGED CONTRACT NUMBER
C
132   NCON=2
      MFLAG=1
139   PAGE=1
      WRITE(5,60) (DATE(I),I=1,20),PAGE
      IF(MSORT.EQ.1) WRITE(6,177) CON(NCON)
      IF(MSORT.EQ.2) WRITE(6,215) CON(NCON)
      IF(MSORT.EQ.13) WRITE(6,211) CON(NCON)
      WRITE(6,11)
      MCOUNT=8
      GO TO 130
C
C              CHANGED CONTRACT NUMBER TO CON(3)
C
138   NCON=3
      MFLAG=2
      GO TO 139
```

```
133   WRITE(6,2101) (DATE(I),I=1,20)
      WRITE(6,2104) CON(PCO)
      DO 3102 K=1,3
      L=YEAR(K)+1900
      IF(L.GE.1900) GO TO 3103
      IF(COUNT(PCO,1,K).LE.0.AND.COUNT(PCO,2,K).LE.0.AND.
     +COUNT(PCO,3,K).LE.0) GO TO 3102
      WRITE(6,2110) TOTAL(PCO,K),(COUNT(PCO,J,K),J=1,3)
      GO TO 3102
3103  IF(COUNT(PCO,1,K).LE.0.AND.COUNT(PCO,2,K).LE.0.AND.
     +COUNT(PCO,3,K).LE.0) GO TO 3102
3111  WRITE(6,2103) L,TOTAL(PCO,K),(COUNT(PCO,J,K),J=1,3)
3102  CONTINUE
      WRITE(6,2107) CON(PCO),GTOTAL(PCO,4),(GTOTAL(PCO,J),J=1,3),
     +PERCNT(PCO)
      IF(PCO.EQ.1) GO TO 132
      IF(PCO.EQ.2) GO TO 130
      IF(MSORT.EQ.13) GO TO 2100
      IF(MSORT.EQ.2) GO TO 135
C
C              SORT ON CATEGORY ORDER PER CONTRACT NUMBER
C
      MSORT=2
      MFLAG=0
      PAGE=1
      PCOP=1
      WRITE(6,50) (DATE(I),I=1,20),PAGE
      WRITE(6,210) CON(PCOP)
210   FORMAT('0',41X,'SORTED BY CATEGORY ORDER PER CONTRACT NUMBER',
     +1X,I4)
      WRITE(6,11)
      PCOUNT=0
184   READ(7,1,END=133) (NONE(I),I=1,12),PRICE,(NONE(J),J=13,35)
      IF(NONE(14).EQ.BLANK) GO TO 5119
      AMP=AMPER
      GO TO 5119
5118  AMP=BLANK
5119  CONTINUE
      SIGN=BLANK
      IF (PRICE.GE.0.0) GO TO 9921
      PRICE=(-1)*PRICE
      SIGN=MINUS
9921  CONTINUE
      IF ((NONE(8).EQ.3D73.AND.MFLAG.EQ.0).OR.(NONE(8).EQ.5079.AND.
     +MFLAG.EQ.1)) GO TO 184
      WRITE(6,7) (NONE(J),I=3,11),(NONE(K),K=1,2),(NONE(L),I=11,12),
     +PRICE,SIGN,NONE(13),AMP,(NONE(KK),KK=14,23),
     +(NONE(LL),LL=24,35)
      PCOUNT=PCOUNT+1
      IF(PCOUNT.GT.50) CALL HEADING(PCOUNT)
      GO TO 184
C
C              SORT ON CALL ORDER PER CONTRACT NUMBER
C
135   MSORT=13
      MFLAG=0
      PAGE=1
      PCOP=1
```

```
      WRITE(6,60) (DATE(I),I=1,20),PAGE
      WRITE(6,211) (CON(M),M=1,6)
211   FORMAT('0',43X,'SOCIETY CHECK NUMBER FOR CONTRACT NUMBER',1X,I4
      WRITE(6,11)
      NCOUNT=0
451   READ(6,1,END=135) (MORE(I),I=1,12),PRICE,(MORE(J),J=13,35)
      IF(MORE(14).EQ.BLANK) GO TO 5120
      ANS=ANSER
      GO TO 5121
5120  ANS=BLANK
5121  CONTINUE
      SIGN=BLANK
      IF (PRICE.GE.0.0) GO TO 9922
      PRICE=(-1)*PRICE
      SIGN=MINUS
9922  CONTINUE
      IF ((MORE(8).EQ.3072.AND.BLANK(13,5).OR.(MORE(6).EQ.5079.AND.
     +BLANK.EQ.1)) GO TO 175
      WRITE(6,9) (MORE(J),J=2,10),(MORE(K),K=1,2),(MORE(L),L=11,12),
     +PRICE,SIGN,MORE(13),ANS,(MORE(KK),KK=14,21),
     +(MORE(LL),LL=24,35)
      NCOUNT=NCOUNT+1
      IF(NCOUNT.GT.59) CALL NEWPAG(NCOUNT)
      GO TO 451
2100  WRITE(6,7000)
7000  FORMAT('1')
      STOP
      END
```

DATE: November 1, 1972

PROJECT: Inventory Updates

Attachment 3

CARD LAYOUT FORM

Type of Card Input Update Cards

Field No.	Field Description	Columns From	Columns Thru	No. of Columns	Coding Instructions	
1	Local Id.	1	5	5	Right Justify	Numeric
2		6	6	1	Code a hyphen	
3	Category	7	8	2	Right Justify	Numeric
4	Article	9	25	17	Left Justify	Alphanumeric
5	Contract Number	26	34	9	Left Justify	Alphanumeric
6	Adm-prog-sp. prog.	35	35	1		Alphabetic
7	Budget Year	36	37	2		Numeric
8	Unit Price	38	46	9	Right Justify	Numeric (Decimal pt. in Column 44)
9	Check Number 1	47	51	5	Right Justify	Numeric
10		52	52	1	Code a & if there are two check numbers coded	Alphanumeric
11	Check Number 2	53	57	5	Right Justify	Numeric
12	Date	58	62	5	Right Justify	Numeric/Numeric*
13	Serial Number	63	76	14	Left Justify	Alphanumeric
14		77	78	2	Blank	
15	Update Code	79	79	1	Code N - Insert, C - Change, or D - Delete	
16	Card Number	80	80	1	Code a "1"	
	CARD 2					
1	Local Id.	1	5	5	Right Justify	Numeric
2		6	6	1	Code a hyphen	
3	Category	7	8	2	Right Justify	Numeric
4	Description	9	46	38	Left Justify	Alphanumeric
5	Location	47	54	8	Left Justify	Alphanumeric
6		55	78	24	Blank	
7	Update Code	79	79	1	Code N - Insert, C - Change, or D - Delete	
8	Card Number	80	80	1	Code a "2"	

Comments:

DISTRIBUTION LIST

Accounting

Patty Carpenter

Lanaii Kline (2)

Frank Teplitzky

Directorate

Library & Archives

Bill Coulton